REQUIREMENT DOCUMENTATION GUIDELINES

1.0 REQUIREMENTS

The purpose of the requirements is to focus upon the external software product theat the user see (i.e., what functions are to be performed on what data to produce what results. It describes the logical relationships among functions and data, and may include design constraints.

Each requirement must be testable;(i.e., if the user is asked for a demonstration of this requirement, a test could be formulated and run showing that the system meets the requirement. This means that you must be careful how you write the requirements because you can make the testing process easier or harder.

Requirements numbering is useful if the requirements are going to be baselined and tracked and cross referenced to other documents and steps in the development process.

The Requirements Document should describe the product in the following terms, as applicable:

External Interfaces - description of inputs and outputs from the software

User Interface - description of the logical characteristics of each interface between the software product and the user. Types of screens, commands and reports, input/output devices. Includes types of messages that are displayed acknowledged and logged.

Hardware Interface - description of the logical characteristics of each interface between the hardware components of the system and software.

Software Interfaces - describes uses of other required software products or interfaces. This may include database systems operating systems and or mathematical packages

Communication Interfaces - describes protocols for communication with entities such as a local area network

Functional Specifications - Actions that take place in the software to accept and process inputs as well as process and gather outputs. This includes validity checks, algorithms sequences of operations, abnormal situation responses, effect of parameters and relationships of outputs to inputs (sequences, conversions). Data Flow Diagrams or class diagrams may be used to show the logical relationships, but not the design.

Performance - specify the static and dynamic numerical requirement placed on the whole software product or upon human interaction with the whole product. Static numerical requirements include terminals support, number of users, amount/type of information handled. Dynamic numerical requirements included transactions/tasks/data to be processed with in time periods for normal and peak workload conditions

Logical Database requirements - list the requirement for information that is to be placed in a database. This includes types of information, frequency of use, accessing capabilities, data entities and their relationships, and retention requirements

Design constraints - Language, standards regulations and hardware limitations

System Attributes - Reliability, availability, security, maintainability and portability

2.0 INSPECTION CHECKLIST FOR REQUIREMENTS DOCUMENT

Organization and Completeness

- ☐ Has each major function been adequately specified?
- ☐ Are all external hardware, software, and communication interfaces defined?
- ☐ Have algorithms intrinsic to the functional requirements been defined?
- ☐ Have the database requirements been specified?
- ☐ Are any design constraints specified?
- ☐ Are all internal cross-references to other requirements correct?
- ☐ Are all requirements written at a consistent and appropriate level of detail?
- ☐ Do the requirements provide an adequate basis for design?
- ☐ Does the specification include all of the known customer or system needs?
- ☐ Is any necessary information missing from a requirement? If so, is it identified as TBD?
- ☐ Is the expected behavior documented for all anticipated error conditions?

Correctness

- ☐ Do any requirements conflict with or duplicate other requirements?
- ☐ Is each requirement written in clear, concise, unambiguous language?
- ☐ Do requirements answer the question “what” not “how”?
- ☐ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ☐ Is each requirement in scope for the project?

- ☐ Is each requirement free from content and grammatical errors?
- ☐ Can all of the requirements be implemented within known constraints?

Quality Attributes

- ☐ Are all performance objectives properly specified?
- ☐ Are all reliability, constraints, and safety considerations properly specified?
- ☐ Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?

Traceability

- ☐ Is each requirement uniquely and correctly identified?
- ☐ Is each requirement traceable to a higher-level requirement (e.g., system requirement, use case)?

Special Issues

- ☐ Are all requirements actually requirements, not design or implementation solutions?
- ☐ Are the time-critical functions identified, and timing criteria specified for them?